A Work Project, presented as part of the requirements for the Award of a Masters Degree in Finance from *Nova School of Business and Economics*

PREDICTING MARKET DIRECTION WITH HIDDEN MARKOV MODELS

ARTUR PEDRO ANTUNES DA SILVA

#694

A directed research project carried out with the supervision of:

Professor Pedro Lameira

January 2014

Abstract

This paper develops the model of Bicego, Grosso, and Otranto (2008) and applies Hidden Markov Models to predict market direction. The paper draws an analogy between financial markets and speech recognition, seeking inspiration from the latter to solve common issues in quantitative investing. Whereas previous works focus mostly on very complex modifications of the original hidden markov model algorithm, the current paper provides an innovative methodology by drawing inspiration from thoroughly tested, yet simple, speech recognition methodologies.

By grouping returns into sequences, Hidden Markov Models can then predict market direction the same way they are used to identify phonemes in speech recognition. The model proves highly successful in identifying market direction but fails to consistently identify whether a trend is in place. All in all, the current paper seeks to bridge the gap between speech recognition and quantitative finance and, even though the model is not fully successful, several refinements are suggested and the room for improvement is significant.

Keywords: Hidden Markov Models, Speech Recognition, Trading Strategy

1. Introduction and literature review

Individuals have been trying to navigate through the maze of the financial markets for as long as they were first created. Ulrike Malmendier[1] argues that shares have been traded as far back as to the Roman Empire while others point to more recent developments, such as the creation of the Dutch East India Corporation in 1602. While there seems to be little consensus regarding the origin of financial markets, few can argue against the fact that, ever since that day, many have endeavoured to find their way through its maze. With the rise of the computer, investment funds with a quantitative tilt, often named quant funds, rose to prominence. One of the issues with implementing quantitative strategies is adapting to the nonstationarity and, consequently, the changing dynamics of the market and economic environment. In other words, strategies that may prove profitable in one regime may crumble when a change occurs. Evidence of such dynamics has been shown in multiple papers in the literature. The most recognized example is Hamilton (1989), in which the first steps were taken towards modelling regime changes in GNP. Following the seminal work by Hamilton, others followed and regime-switching models have been applied to financial variables such as interest rates in Gray (1996) and volatility in Pagan and Schwert (1990).

In a mostly different but yet parallel world, we have speech recognition. The role of a speech recognition model is to translate speech utterances into written text. These utterances are often highly non-stationary and, as a result, understanding how speech recognition deals with such issues could prove very useful in finance. To tackle this issue, several models such as Dynamic Time Warping (see Myers and Rabiner

[1] The Origins of Value. The Financial Innovations that Created Modern Capital Markets (pp. 31-42, 361-365). Oxford University Press.

(1991)) or Neural Networks (see Graves, Mohamed, and Hinton (2013)) have been proposed.

Nonetheless, there is one model that has proved superior to all others, the **Hidden Markov Model** (HMM). Due to their scalability and overall strong performance, HMMs are the most commonly used model for speech recognition. Through identifying the underlying regimes, HMMs deal with non-stationarity by establishing **state-conditioned stationarity**. Briefly, an HMM is a model with Markov properties built to recognize sequential data, in which a hidden stochastic process is being modelled. The model will be further explained later but a simple way to understand the process is by thinking of your favourite football team. Imagine that, 20 years from now, your memory capacity is lacking and, as a result, you are unable to recall the outcome of every single game your team played. While such is information is not on your diary you do keep a very detailed ranking of how your mood was at the end of each day, ranging from 0(in a very bad mood) to 10(in a great mood). If you filter those days in which there were no games, you are left with a very detailed analysis of how your mood was at the end of each game. To decode the set of game results, one simple approach is to create a threshold model. For example, assume that all games in which a mood above 5 was reported were wins. This seems and probably is a very random and ineffective way to tackle the issue but, fortunately, HMMs excel in such situations. Consider three hidden states, which we assume to be win, draw and loss. By assigning a probability matrix to both the eleven different mood levels and the three different states, HMMs are able to output the optimal state sequence (the game outcomes) by maximizing the probability that the sequence was produced by the model. While the HMM is unable to tell you which state belongs to each outcome, you can

infer that from state characteristics such as mean and standard deviation. It would be logical to assume that the state with the highest mean mood would correspond to a win, the middle one to a draw and the last one to a loss.

As shown, a parallel can be established between the problems of speech recognition and those of finance and, consequently, the HMM is a very attractive solution to deal with financial data. While, as we have previously mentioned, Hamilton (1989) laid the first brick in using a regime-switching approach to model economic variables, Rydén, Teräsvirta and Åsbrink (1998) provided an important contribute by proving HMMs successfully describe stylized facts of price returns. Regarding application on asset allocation, in Ang and Bekaert (2003) a Hamilton inspired regime-switching model is applied to invest in a global asset allocation model in which cash and six equity markets are available and in a market timing model, which allows for an investment in US equity, bonds or cash. They show that there are benefits to the regime-switching approach and Kritzman, Page and Turkington (2012) show similar conclusions but using a different approach. Using an HMM, they model economic variables and not returns directly, to infer whether an event is in place. They apply continuous two-state HMMs to turbulence (Chow, Jacquier, Kritzman and Lowry (1999)), inflation and economic growth.

Another very interesting approach and the one followed in this paper is introduced in Bicego, Grosso and Otranto (2009). The authors chose to apply a discrete HMM[2] instead, to avoid choosing a return distribution. Also, as commonly used to classify words belonging to a finite dictionary, sequences are classified and attributed to

[2] Discrete HMMs differ from continuous HMMs in the variable they observe. Whereas discrete HMMs observe discrete variables such as words in a finite dictionary, continuous HMMs observe continuous variables like financial returns and, consequently, the observation matrix takes the form of a probability density function of some distribution family. Consequently, continuous HMMs output continuous symbols, whereas discrete HMMs output symbols from a discrete dictionary.

an individual HMM, resulting in an HMM for each word. In the paper, the authors apply this approach by discretizing returns (passed on as one if negative and two if positive) and assigning them to one model if the last *f* observations are of increase and to another if they are of decrease. This approach is highly successful and allows for increase and negative sequences to be treated differently, which is a highly documented fact in the literature as pointed out by the paper. While the authors apply this strategy to the Dow Jones Index and other individual stocks, this paper applies the methodology to fifty different strategies that can be traded both long and short. The methodology also differs in the sense that a longer observation period is used, and mixed sequences, those that are neither of increase nor decrease, are tackled differently.

This first section establishes a parallel between speech recognition and the financial world, ultimately proposing that the tool that has been so successfully applied in one sector also be applied in the other. Also, the literature and various applications of the HMM in finance are described, along with a short description of the chosen approach to modelling financial time series data. The remainder of the paper is organized in the following way: in section two the HMM methodology is further explained. In section three, the paper methodology is presented and is divided in first explaining how the fifty strategies were chosen and then in how the HMM is applied to predict market direction. Section four analyses the results and sensitivity to different parameters, while section five concludes.

2. HMM Methodology

HMMs have been used since the late 1960s, mostly due to advances made possible by Leonard Esau Baum and his colleagues at the Institute for Defense Analyses. See for example Baum and Petrie (1966). After an influential tutorial by

Lawrence Rabiner (Rabiner (1989)), HMMs gained widespread notoriety and quickly became the custom tool for a wide variety of speech recognitions problems. The following explanation is inspired by Rabiner's seminal tutorial.

To understand HMMs, one must first comprehend markov chains. In a markov chain, a system has a set of states S, composed of N distinct states. At discrete steps in time, the system can undergo changes of state or stay in the same state. These states changes are modelled by a state-transition matrix A:

$$A = a_{ij}, \forall\, i,j\ \in S\,;\ \sum_{j} aij = 1, i \in S \tag{1}$$

Regarding the initial state, it is modelled by a probability vector π. In notation, markov chains are a 3 tuple $\varphi = (S, A, \pi)$.

For a n-th order markov chain, we have a stochastic process in which the probability of getting into the next states depends only on the n last states. Furthermore, we also assume transition probabilities are time-homogenous. For a first order markov chain, the following conditions are fulfilled:

$$P(X_{t+1} = i|X_t = s_t, X_{t-1} = s_{t-1}, \dots, X_n = s_n) = P(X_{t+1} = i|X_{t-1} = j)\ \forall\, i,j \in S \tag{2}$$

$$a_{ij} = P(X_{t+1} = i|X_t = j) = P(X_t = i|X_{t-1} = j), \forall\, t\ \in T, \forall\, i,j\ \in S \tag{3}$$

Following up on equations 2 and 3, we can easily calculate the probability of a sequence of states, *S*. Using the previous example of the football fan, imagine the following state set, *S*, and transition matrix *A*:

$$S = \{S1, S2, S3\} = \{Win, Loss, Draw\} \tag{4}$$

A	Win	Loss	Draw
Win	0.60	0.15	0.25
Loss	0.20	0.50	0.30
Draw	0.30	0.30	0.40

Given an observation sequence *O = {S1, S1, S3, S1, S2, S2}*, the probability that the sequence was computed by the 3-tuple φ is:

$$P(O|\varphi) = P(S_1, S_1, S_3, S_1, S_2, S_2|\,\varphi\,)$$
$$= \pi * P(S_1|S_1) * P(S_3|S_1) * P(S_1|S_3) * P(S_2|S_1) * P(S_2|S_2) =$$
$$= 1 * 0.60 * 0.25 * 0.30 * 0.15 * 0.50 \sim 0.03$$

Equipped with the knowledge of markov chains, we can now understand HMMs. Whereas in a markov chain the states were visible, S is now hidden. Furthermore, each state emits an observation O_k with a certain probability. This property adds an extra stochastic process and, as a result, HMMs are often defined as double stochastic processes. The observation set, *O*, is visible and the emission of each observation at time t depends solely on the current state. The observation probability matrix is defined as *B* and the probability of O_k given S_j is $b_j(k)$. Therefore, we can define HMMs as the 5-tuple λ = *{π, S, A, O, B}*.

Given the definition of an HMM and our knowledge of markov chains, there are now three questions we must solve to apply it to the financial markets:

1. Given an observation sequence *O*, how do we calculate the probability that such sequence was emitted by the model λ? In other words, how can $P(O|\lambda)$ be computed?

2. Given observation sequence *O*, how do we compute the state sequence S that, according to some criteria, is perceived as being optimal?

3. Lastly, how do we solve for the model parameters λ that maximize the probability, $P(O|\lambda)$, that *O* was emitted by the model?

To solve **problem 1**, we first define $P(O|S,\lambda)$ for a fixed state sequence S:

$$P(O|S,\lambda) = b_{s_1}(O_1) * b_{s_2}(O_2) * \dots * b_{s_t}(O_t) \quad (5)$$

Then, as previously done, we derive the probability of such state sequence, $P(S|\lambda) = \pi_1 * a_{s_1 s_2} * a_{s_2 s_3} * \dots * a_{s_{t-1} s_t}$. The joint probability of *O* and *S*, $P(O,S|\lambda)$, is simply the product of the two terms mentioned above. Then, to get $P(O|\lambda)$ all we have to do iterate $P(O,S|\lambda)$ over all combinations of S. In notation, $P(O|\lambda) = \sum_{all\ S} P(S|\lambda) * P(O|S,\lambda)$.

However, given the need to iterate over all possible state sequences the computing complexity is close to $2T*N^T$ (N is the number of states). This is unfeasible for even small problems but fortunately there is a much more efficient way to compute $P(O|\lambda)$.The technique used is the forward procedure and is defined as follows:

1. **Initiation :** $\alpha_1(i) = \pi_i\, b_i(O_1), \forall\ i\ \in N$
2. **Induction :** $\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) * a_{ij}\right] b_j(O_{t+1})$

   $$j\ \in \{1,\dots,N\}, t\ \in \{1,\dots,T-1\}$$
3. **Termination:** $P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$

We initialize the forward-probabilities as a function of the initial state probability matrix and the first observation. In step 2, the expression between square brackets is essentially the probability of observation sequence O, given that we are in state j at t+1. By multiplying the expression by $b_j(O_1)$ we account for the probability of obtaining emission O_{t+1} given state j. Hence, α can be defined as the probability of

observation sequence O while being in state j,$P(O, S_j|\lambda)$. To compute $P(O|\lambda)$ we need only to sum α over N.

We will also take the opportunity to define the backward variable, *β.* It is closely related to the forward variable *α* and **while not needed to solve the current problem,** problem 3 requires its usage and, given the similarity to *α*, now is the right time to introduce such concept. The forward probability,$\beta_t(i)$, can be defined as the probability of the partial sequence starting at *t+1* and ending at *T*, given the model *λ* and that at time *t* it is in state *i*.

$$\beta_t(i) = P(O_{t+1}O_{t+1} \dots O_T|q_t = S_i, \lambda)$$

1. **Initiation :** $\beta_T(i) = 1, \forall\, i \in N$
2. **Induction :** $\beta_t(i) = \sum_{j=1}^{N} a_{ij}\, b_j(O_{t+1})\beta_{t+1}(i)$

$$i \in \{1, \dots, N\}, t \in \{T-1, T-2, \dots, 1\}$$

The logic behind the procedure is the inverse of that used for the forward method. The method is initialized at point *T*, the end of the sequence. In order to be in state *i* at time *t* and to take in consideration the observation sequence starting at time *t+1*, one must account for all possible states *j* at *t+1* and for the probability of observing O_{t+1}while in state *j.* By also accounting for the remaining partial sequence from state *j*, as included in the definition of *β,* we have step 2. Again, we will leave the application of this concept to problem 3.

Onto **problem 2**, this is known as the decoding problem, since the goal is to decode the state sequence that maximizes a given criteria. Consequently, the first issue is choosing the optimality criteria. One can choose at each point in time what the individually most likely state is, by simply choosing the state that emits the current observation with the highest probability. Nonetheless, the most widely used criterion is

choosing the state sequence that maximizes the probability of emitting observation sequence O. This can be defined as maximizing $P(S|O,\lambda)$ and is achieved by a dynamic programming technique called the **Viterbi algorithm** (Viterbi (1967)). To understand the Viterbi algorithm we start by defining δ:

$$\delta_t(i) = max_{s_1,s_2,\dots,s_{t-1}} P(s_1 s_2 \dots s_t = i, O_1 O_2 \dots O_t|\lambda) \quad (7)$$

The variable δ can thus be defined as the highest probability score for a given state set *S* and the partial observation *O* ending at time *t* and state *i*. Inductively we can define $\delta_{t+1}(j) = \left[max_i\, \delta_t(i) * a_{ij}\right] * b_j(O_{t+1})$.

To obtain the optimal sequence we must keep a log of the states that, for each *t* and *j*, maximize equation N. This log is kept on array $\psi_t(i)$ and the procedure is as follows:

1. **Initialization :**

   $\delta_t(i) = \pi_i b_i(O_1), \qquad 1 \le i \le N$

   $\psi_1(i) = 0$

2. **Recursion :**

   $\delta_t(j) = max_{1\le i\le N}\left[\delta_{t-1}(i) * a_{ij}\right] b_j(O_t),$

   $2 \le t \le T, 1 \le j \le N$

   $\psi_t(j) = argmax_{1\le i\le N}\left[\delta_{t-1}(i) * a_{ij}\right],$

   $2 \le t \le T, 1 \le j \le N$

3. **Termination :**

   $P^* = max_{1\le i\le N}[\delta_t(i)]$

   $S_t = argmax_{1\le i\le N}[\delta_t(i)]$

4. **Backtracking :**

   $S_t^* = \psi_{t+1}(S_{t+1}^*), \qquad t = T-1, T-2, \dots, 1$

The Viterbi Algorithm, except for the last step, is identical to the Forward Procedure, with the exception that only the maximum value is recorded. The Viterbi Algorithm finds the optimal state sequence given an observation sequence *O* by keeping track of the argument that maximizes the probability along a single path, for each *j* and *t*.

With only **problem 3** left, we now have to deal with the most challenging issue, how to adjust the parameters λ in order to maximize the probability that sequence *O* was produced by λ, $P(O|\lambda)$. The difficulty arises mostly due to the fact that for any given sequence, there is no optimal way of estimating the model parameters or, in other words, only local *maxima* can be computed. The method chosen to tackle such issue is an iterative procedure named **Baum-Welch**, which does so by only stopping searching for the optimal parameters once a set maximum number of iterations is achieved or the improvement on the sequence's likelihood is below a given threshold. The procedure relies mostly on the forward and backward variables we have defined in step 1.

Firstly, we will define $\xi_t(i,j)$, as the probability of observing state *i* at time *t* and state *j* at time *t+1*, given observation sequence *O* and model λ. In notation:

$$\xi_t(i,j) = P\left(s_t = S_i, s_{t+1} = S_j \,|O,\lambda\right) \tag{8}$$

Through Bayes rule and our previous knowledge of the forward and backward procedure, we can define $\xi_t(i,j)$ as follows:

$$\xi_t(i,j) = \frac{P(q_t = S_i, q_{t+1} = S_j, O|\,\lambda)}{P(O|\,\lambda)} = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\,\lambda)} \tag{9}$$

In equation 9, $\alpha_t(i)$ accounts for the partial sequence until *t*, $a_{ij}b_j(O_{t+1})$ accounts for the transition to state *j* and occurrence O_{t+1} given that we are in state *j*, while $\beta_{t+1}(j)$ accounts for the partial sequence from *t+1* until *T*.

Hence, it is straightforward to understand that by summing ξ_t over *T-1,* we can infer the expected number of transitions from state *i* to *j*. The reason why we sum only until T-1 is that, by definition, no transition is done at T. An interesting next step would be to define the expected number of transitions from state *i*. We will first define the probability of being in state *i at* time *t* given observation sequence *O* as:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} \tag{10}$$

Following the previous reasoning, if one sums γ_t over *T-1*, we get the expected number of transitions from state *i*. Summarizing all this information, we now have:

$$\sum_{t=1}^{T-1} \gamma_t(i) = Expected\ number\ of\ transitions\ from\ state\ i \tag{11}$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = Expected\ number\ of\ transitions\ from\ state\ i\ \ to\ state\ j \tag{12}$$

Now, for the final step, we can use the previous two definitions, to calculate the values of the initial distribution, the transition probability matrix and the observation matrix. They are defined as follows:

$$\pi_i = \gamma_1(i), \qquad 1 \leq i \leq N \tag{13}$$

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t}^{T-1} \gamma_t(i)} \tag{14}$$

$$b_j(k) = \frac{\sum_{t_{s.to}\, o_t=k}^{T-1} \gamma_t(j)}{\sum_{t}^{T} \gamma_t(j)} \tag{15}$$

With this information, we are now equipped to compute the HMM's parameters through the Baum-Welch algorithm. Before moving on to the next chapter, in which the different strategies created and the motivation behind them are explained, some remarks are due. Firstly, one important parameter is the number of states. In practice, there is no single answer for such a problem. One solution is to choose a number that is logical

given the problem, such as for the football fan case in which three states were chosen, or to choose the number of states according to a statistical criterion. The two main criteria are the Bayesian information criterion (Schwarz (1978)) and the **Akaike information criterion (AIC)** from Akaike (1973), which will be the one implemented in this work. Also, concerning the computation of the sequence's likelihood, given the usually small number of the values for *a* and *b*, as *t* increases the values to be summed quickly exceed the precision range of any computer. To avoid such situation, the logarithm of the likelihood is taken instead. The necessary changes are very little and for more information please see Rabiner (1989). Also, as we have previously mentioned, only local *maxima* can be found and, consequently, the initiation procedure is very important. In this work we tackle this issue by randomly initializing the model, running it several times and recording the parameters that produce the highest likelihood. In Rabiner (1989), more complex solutions such as segmentation algorithms are discussed in more depth. Furthermore, while in this work we assume all states can be reached from any state, other more complex architectures, such as defining a terminal state in which all transitions end, exist. Lastly, whereas our focus is in discrete HMMs, they can also be applied to continuous data. In that case, the observation probability matrix is instead a function of a given probability distribution. Again, and the same applies for a thorough introduction to more HMM architectures, please refer to Rabiner (1989).

3. Model implementation and result analysis

While HMMs have been extensively used in financial applications, their usage is mostly confined to asset allocation problems. In the light of Bicego, Grosso and Otranto (2009), we chose to apply HMMs to predict market direction. Moreover, while some applications to active trading strategies exist (see Zhang (2001)), they mostly rely on

complex modifications of the existing algorithm. **In this paper we chose to use the standard HMM algorithm but in an innovative way, by seeking inspiration from speech recognition**. While the core of our methodology is derived from Bicego, Grosso and Otranto (2009), we differ mostly from their approach by applying HMMs to a set of strategies and not just one asset individually. This is significant since, given the different characteristics of each strategy, we will understand how the HMM tackles different return characteristics. Furthermore, as it will be further explained, trade sequences are divided in those that trend and those that do not. Alternative measures to deal with the latter are introduced through an analogy to speech recognition problems. Lastly, by using data for the S&P500 Index since 1950 until October 2014, our results present improved significance. Summarizing, **our methodology will consist first in creating the base investment strategies and then in creating HMMs to predict the return signals for each strategy independently**.

The reasoning behind creating a set of strategies and not only one, derives from the fact that when developing a quantitative strategy one of the main concerns is that the in-sample success is mostly due to **over fitting** the strategy to the training set and that, once the strategy is tested out-of-sample, it will fail. If the HMM performs consistently across parameters it is fair to assert that our over fitting concerns are eased.

Our reasoning is best understood by resorting again to the concept of regimes. Quantitative strategies are commonly inspired by one of two core market regimes, **mean reversion**[3] and **trend following**[4]. When in a mean reverting regime, asset prices return to the mean after deviating from it, while when in a trend following regime asset

[3] See Bali and Demirtas (2006) for an application of mean reversion modelling to stock volatility and Huang, Jiang, Tu and Zhou (2013) for a broader approach

[4] See Clare, Seaton, Smith and Thomas (2006) for an application of trend-following strategies to the commodity market and Antonacci (2014) for an application of the strategy to a broader portfolio of assets.

prices follow a trend, either up or down. Thus, we will define trend following strategies as those in which we buy when price rises for *l* consecutive days and sell when it drops for *l* consecutive days. On the other hand, mean reverting strategies are those in which we buy when price decreases for *l* consecutive days and sell when it increases for *l* consecutive days. Over the look back period *l*, for both the mean reverting and the trend following strategy sets, we will consider all possible buy and sell combinations. For a better understanding, table 1 contains a sample of some entry and exit combinations for both strategy sets.

Table 1 – Sample combinations for the Trend Following and Mean Reversion strategy sets

Trend Following set	
Buy if price increases for N consecutive days	**Sell if price decreases for N consecutive days**
1	1
2	1
3	1
4	1
5	1

Mean Reversion set	
Buy if price decreases for N consecutive days	**Sell if price increases for N consecutive days**
1	1
2	1
3	1
4	1
5	1

Throughout the paper we will be using $l = 5$, which results in a strategy total of 50, 25 strategies for the trend following set and other 25 for the mean reversion set. Furthermore, since a strategy can also be traded short there are 100 strategies in practice. A deeper analysis of the performance metrics for both strategy sets is available in table 2 and, for a visual understanding of the very diverse characteristics of each strategy, the annualized returns and volatility are plotted in appendix 1-2.

Table 2 – Performance metrics for both strategy blocks

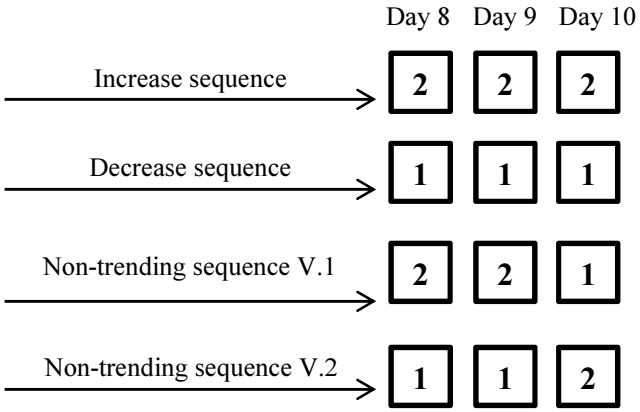
	Trend Following	Mean Reversion
Annualized return	4.72%	2.62%
Annualized volatility	7.98%	9.31%
Correlation to the market	87.35%	90.87%
% of positive days	51.55	50.32
% of positive months	54.74	64.23
Kurtosis	133.94	18.26
Skewness	-0.86	-2.59
Maximum Drawdown	17.08%	50.04%

On to the second step, we will now discuss how return signals are predicted for each strategy. Firstly, we will define a sequence as a vector composed by the trade returns[5] of the last *s* days, the **sequence length**. For each sequence, our purpose is to forecast the return signal for the last *fo* days, the **forecast length**. This is achieved by filtering only those sequences that on the last *f* days, the **filter length**, had the same return signal. Hence, we can define the **increase set** as the set of sequences in which the last *f* returns were all positive and the **decrease set** as the set of sequences in which the last *f* returns were all negative. These two sets are defined as **trending.** However, while this is fine for training, when testing our strategy out of sample, the days meant to be forecasted are not present in the sequence and, consequently, there is uncertainty regarding the days meant to be forecasted. This results in the creation of two other sequence sets, that in which from *s-f+1* to *s-fo* returns are positive and then in *s-fo+1* to *s* are negative and that in which from *s-f+1* to *s-fo* returns are negative and then in *s-fo+1* to *s* are positive (in other words, the return signal reverses in the forecast period). These two sequence sets are defined as **non-trending**. For a better understanding of how both non-trending and trending sets are built, figure 1 showcases both sequence

[5] Given that the strategies are not active every day, returns are actually grouped into trades. Hence, each return actually represents the total return over a trade.

types for *s = 10, f=3* and *fo=1.* In the figure, the number 2 represents positive returns whereas the number 1 represents negative returns[6].

Figure 1 – Sequence types for *s = 10, f=3* and *fo=1*



When fully implementing the strategy, data will only be available until day 9. Hence, an investor must not only distinguish between increase and decrease sequences but also filter non-trending sequences, since if day 10 is removed one is unable to distinguish increase sequences from the first version of non-trending sequences and decrease sequences from the second version. While our focus is mostly on the former we will also tackle the latter by, again, drawing inspiration from speech recognition. Succinctly, we will now first explain how the model is trained, then how increase and decrease sequences will be classified given the *a priori* knowledge of which sequences trend and, ultimately, we will tackle ignoring those that do not trend.

The training methodology relies on an analogy to speech recognition[7], in which an HMM is trained for each phoneme type and phonemes are then classified according

[6] In order to implement a discrete HMM, as mentioned, data was discretized by setting positive returns to 2 and all others to 1.
[7] See Gales and Young (2008) for a review of similar applications in speech recognition.

to the HMM that outputs the highest likelihood[8]. **Given the *a priori* knowledge of which sequences trend**, we will then split sequences into those of increase and those of decrease. Ultimately, an HMM will be trained for each of the sequence types, resulting in an **increase HMM** and a **decrease HMM**. Then, all test sequences will be processed by both HMMs and will be classified according to the HMM that produces the highest likelihood. The training procedure uses 50% of the dataset and the remaining portion will be used for out of sample forecasting. Regarding out of sample forecasting, the issue concerns **how to initiate the Baum-Welch procedure and how much data to use at any point in time**. The initiation issue is tackled by using the values derived from training for the first data point and then always use the previously computed parameters to initiate the model. As regards the amount of data used to estimate the parameters, one choice would be to use the same parameters as those derived from training as in Bicego, Grosso and Otranto (2009). However, such option only suffices for small data samples, which is not our case. We chose to instead make a 5-year rolling estimation of the parameters, thus using only trade sequences that happened on the last 5 years. The performance of our model is analysed by measuring the percentage of correctly classified sequences for each parameter combination (table 3) and by the reward to risk ratio [9](RR) as in figure 2[10].

On the one hand, the overall idea is that as the length of the sequence increases the accuracy of the model deteriorates. This is in accordance to the idea that **longer-term patterns are quickly destroyed by the market**. On the other hand, as filter

[8] As previously mentioned, and similarly to speech recognition applications, a discrete HMM will be used. Moreover, the Baum-Welch is randomly initiated as it commonly is in the literature.
[9] The risk reward ratio is defined as the ratio of annualized returns and annualized volatility. It can be interpreted as how many units of return are available per unit of risk.
[10] All results are based on an equal-weighted average of all active strategies in any given day. Moreover, a forecast length of 1 is used for all analyses.

length increases the accuracy improves. This is likely due to the fact that increasing the filter length increases the 'exclusiveness' of the sequence and, consequently, sequences have more differentiating characteristics.

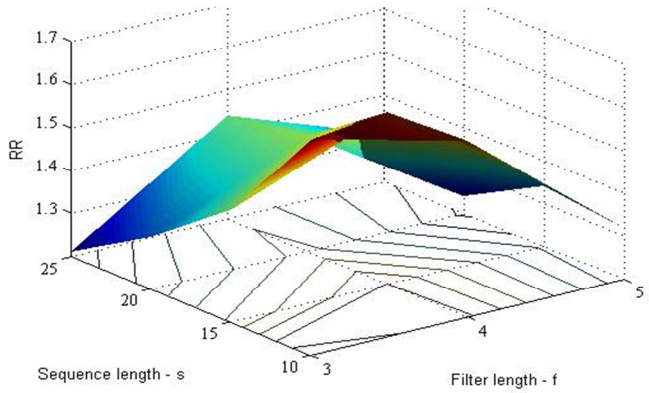
Table 3 – Percentage of correctly classified sequences for both increase and decrease sets

		% Detected correctly	
Sequence length	**Filter length**	**Increase**	**Decrease**
10	3	88.02	87.94
10	4	88.08	92.73
10	5	87.51	97.73
15	3	84.51	78.83
15	4	87.96	90.00
15	5	87.78	94.81
20	3	83.64	73.45
20	4	86.88	86.08
20	5	87.17	91.63
25	3	82.08	68.04
25	4	86.37	82.28
25	5	87.17	91.29

Moreover, it is also interesting to note that the relationships between accuracy and both parameter sets do not translate directly to the RR. As both the sequence length and the filter length increases, the RR decreases. While for the sequence length the reasoning is the same, an increase in filter length now results in a performance decrease due to the lower observation number. Observations decrease greatly as the filter length increases and, consequently, even a small error rate may have a big impact on the RR. This is especially true for our strategies, which have very a reduced amount of observations. This relationship is plotted in appendix 3. The figure contains the relationship between the RR and the threshold for the number of training observations required per strategy. All strategies that have a number of training observations below a given level are excluded from our dataset. The relationship between performance and the observation number is clearly positive, mainly when comparing high to low limits.

This is an interesting starting point for improvement and suggests that by increasing the amount of training data, performance can be greatly improved.

Figure 2 – Sensitivity analysis of the RR to multiple parameter combinations



The same effect applies to the strategy's correlation with the market (appendix 4), as it greatly decreases as the threshold increases. Furthermore, using the best parameter set *s = 10* and *f=3*, table 4 contains several performance metrics. The model performs very well[11] and the fact that all calendar years showed positive returns is striking, as well as the very low drawdown amount. The high correlation value is worrying but, as previously discussed, can be mostly solved by increasing the number of observations used for training.

Table 4 – Performance metrics for parameter set s = 10 and f=3

	HMM strategy	S&P500 Index
Annualized return	25.78%	8.96%
Annualized volatility	14.48%	18.17%
RR	1.78	0.49
Correlation to the market	80.00%	100.00%
% of positive days	52.29	0.54
% of positive months	74.75	0.77
% of positive years	100.00	0.61
Daily kurtosis	12.88	30.59
Daily skewness	0.43	-1.22
Maximum Drawdown	4.14%	22.88%

[11] See appendix 5 for the Profit and Loss plot for both the HMM strategy and the S&P500 Index.

Lastly, not only is the model very robust across parameters (figure 2) but also tackles different strategies in an excellent manner, showing very little difficulty in adapting to different strategies (see appendix 6) and further easing our over fitting concerns.

Now that we have successfully distinguished increase and decrease sequences, it is now time to address the filtering of non-trending sequences. **The focus of this section is to apply the strategy to real-life situations, in which *a priori knowledge* of which sequences trend is non-existent**. As illustrated in figure 1, for the case of *s = 10, f=3* and *fo=1*, two types of non-trending sequences exist. To tackle removing non-trending strategies, three methodologies will be implemented. The first two rely on garbage models[12], which are implemented in speech recognition to ignore out-of-vocabulary words, those not meant to be recognized. For the first garbage model (garbage model 1) one HMM will be trained for all the non-trending sequences, whereas for the second garbage model (garbage model 2) one HMM is trained for each type of non-trending sequence. The last methodology (pdf garbage model) is inspired by Bicego, Grosso and Otranto (2009) and is best understood by defining the concept of confidence Θ:

$$\Theta = |likelihood(increase) - likelihood(decrease)|$$

For those sequences in which Θ is below a threshold ϵ, the sequence is deemed as non-trending. The value ϵ is defined by finding the interception of the ϵ probability density functions for both trending and non-trending sequences. This is as perceived as the point of least error and an example is available in appendix 7.

Summarizing, for the two garbage model methodologies, both non-trending and trending sequences will be processed through the increase HMM, the decrease HMM

[12] Inspired by Dunnachie, Shields, Crawford and Davies (2009), in which HMM garbage models are tested against other methodologies and compare very favourably.

and the garbage model. If the garbage model outputs the highest likelihood the sequence will be classified as non-trending. Regarding the probability density function approach, both non-trending and trending sequences will be processed through the increase and decrease HMM. All sequences for which Θ is below ϵ will be classified as non-trending.

By incorporating the usage of methodologies to filter non-trending sequences, we are now fully equipped to apply our model to real-life situations. **The goal is to process both non-trending and trending sequences through the model and then ignore those that do not trend and correctly predict the signal for those that trend.** For an analysis of the accuracy of all methodologies, the confusion matrices are available in tables 5.1-3. Furthermore, performance metrics for all the filtering procedures are available in appendix 8.

Table 5.1 – Confusion matrix for garbage model 2

		Predicted		
		Increase	**Decrease**	**Non-trending**
Target	**Increase**	2287(62%)	0(0%)	1373(38%)
	Decrease	0(0%)	1123(60%)	744(40%)
	Non-trending	1281(33%)	961(25%)	1615(42%)

Table 5.2 – Confusion matrix for garbage model 1

		Predicted		
		Increase	**Decrease**	**Non-trending**
Target	**Increase**	2232(61%)	0(0%)	1428(39%)
	Decrease	0(0%)	1119(60%)	748(40%)
	Non-trending	1277(33%)	1017(26%)	1563(41%)

Table 5.3 – Confusion matrix for pdf garbage model

		Predicted		
		Increase	**Decrease**	**Non-trending**
Target	**Increase**	1729(47%)	0(0%)	1931(53%)
	Decrease	0(0%)	831(45%)	1036(55%)
	Non-trending	976(25%)	792(21%)	2089(54%)

All three methodologies are unable to correctly identify non-trending sequences. Not only do they misclassify non-trending sequences but also classify trending sequences as non-trending. This suggests that either trending and non-trending sequences are too much alike or the HMM is not being able to adjust to the complexity of the data. If the former is the case, derivations of the HMM to adjust for multiple observation sequences could be useful, whereas for the latter more complex HMM algorithms such as Hierarchical Hidden Markov Models (HHMMs) or HMMs with time-varying transition matrices may be better suited. The complexity reasoning loses

plausibility given that, when choosing the optimal number of states through the AIC, all sequence sets show a low and very similar amount of states[13].

4. Conclusion

Speech Recognition provides an important stepping stone to tackle some of finance's toughest problems. The enormous success of HMMs in speech recognition architectures and the model's ability to tackle sequential data, result in an excellent candidate for quantitative investment solutions.

Throughout the paper a basic overview of HMM theory is provided along with its implementation to market direction forecasting in the S&P500 Index. Whereas the few applications of HMMs to active quantitative strategies that exist focus mostly on complex modifications of the standard algorithm, the current paper focuses on developing an analogy to speech recognition and applying the seasoned HMM algorithm, which has proved so successful throughout the years.

The current approach also solidifies the algorithm's potential by showcasing its robustness out of sample and across different strategies. The model is tested against 50 different strategies and exhibits a very consistent behaviour. Furthermore, some important conclusions can also be drawn from our analysis. As expected, longer term sequences seem to be less predictable as market participants quickly erode any edge. The model's sensitivity to the amount of data observations is also worthy of note, as it can be an interesting starting point for further improvement. While the chosen strategies provide an important benchmark due to their diversity, in practical implementations the model could also be applied to proprietary models. However, the model fails short in predicting which sequences trend and which do not. Despite implementing multiple

[13] See appendix 9 for a comparison between non-trending and trending sequences for the average number of states.

garbage models to filter out non-trending sequences, we are unable to correctly distinguish both sequence types. The inadequacy is constant across sequence types and is either caused by the model's failure to tackle the complexity of the data or the sequences simply do not possess enough differentiating characteristics. For the former, as previously mentioned, one could implement HHMMs or a time-varying transition matrix[14]. HHMMs allow for a more complex portrayal of the market's dynamic by making each of its states an independent probabilistic model, while HMMs with time-varying transition matrices allow for the observed variable to be influenced by other covariates or even its lagged values. As for the latter, methodologies such as a Dynamic Naïve Bayes classifier (see Avillez-Arriaga, Sucar and Mendoza (2006)) allow for a HMM-like approach to modelling multiple observed variables. Summarizing, further refinements are possible and provide plenty of room for improvement

We conclude by summarizing the importance of the relationship between speech recognition and finance. The recent success and rise of speech recognition models is of great interest to finance practitioners and, given the model's place at the front of the speech revolution, it is only logical that HMM methodologies could prove very useful. While the model is not fully successful, by bridging the gap between speech recognition and quantitative investing, the current paper provides an important stepping stone towards further development. Moreover, despite a seemingly oversimplifying approach to data modelling, HMMs are able to tackle financial data in a very robust and highly efficient matter. All in all, **by dealing with non stationarity in a simple yet eloquent way HMM based strategies are an alluring alternative**.

[14] See Fine, Singer and Tishby (1998) for an introduction to HHMMs and see Meligkotsidou and Dellaportas (2011) for an application of time-varying transition matrices with HMMs

5. References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*,19: 716-723.

Ang,A., & Bekaert, G. (1990). How do Regimes Affect Asset Allocation (Working Paper No.10080). Retrieved from National Bureau of Economic Research website http://www.nber.org/papers/w10080.pdf.

Antonacci, G. (2014). Absolute Momentum: A Simple Rule-Based Strategy and Universal Trend-Following Overlay. *Portfolio Management Consultants*.

Avilés-Arriaga, H., Sucar, L., & Mendoza, C. (2006).Visual Recognition of Similar Gestures. *International Conference on Signal Recognition*, 1: 1100-1103.

Bali, T., & Demirtas, K. (2006).Testing Mean Reversion in Stock Market Volatility.*Journal of Futures Markets, 28(1)*: 1-33.

Baum, L., & Petrie, T. (1966). Statistical Inference for Probabilistic Functions of Finite State Markov Chains.*The Annals of Mathematical Statistics, 37(6)*: 1554-1563.

Bicego, M., Grosso, E., & Otranto, E. (2008). A Hidden Markov Model Approach to Classify and Predict the Sign of Financial Local Trends. Sassari, Italy: DEIR - University of Sassari.

Chow, G., Jacquier, E., Kritzman, M., & Lowry, K. (1999). Optimal Portfolios in Good Times and Bad. *Financial Analysts Journal*,55(3): 65-73.

Clare, A., Seaton, J., Smith, P. & Turkington, D. (2012). Trend Following, Risk Parity and Momentum in Commodity Futures. (Discussion paper, Department of Economics, University of York).

Dunnachie, M., Shields, P., Crawford, D., & Davies, M. (2009). Filler Models for automatic speech recognition created from hidden Markov models using the K-Means algorithm. *European Signal Processing Conference*, 17: 544-548.

Fine, S., Singer, Y., & Tishby, N. (1998).The Hierarchical Hidden Markov Model: Analysis and Applications. *Mahcine Learning, 32*: 45-62.

Gales, M., & Young, S. (2007). The Application of Hidden Markov Models in Speech Recognition. *Foundations and Trends in Signal Processing, 1(3)*: 195-304.

Graves, A., Mohamed, A., & Hinton, G. (2013). Speech Recognition with Deep Recurrent Neural Networks. *International Conference on Acoustic Speech and Signal Processing*. Vancouver, Canada.

Gray, S. (1996). Modeling the conditional distribution of interest rates as a regime-switching process. *Journal of Financial Economics*, 42: 27-62.

Hamilton, J. (1981). A New Approach to the Economic Analysis of Nonstationarity Time Series and the Business Cycle. *Econometrica*, 57(2): 357-384.

Huan, D., Jiang, F., Tu, J., & Zhou, G. (2013). Mean Reversion, Momentum and Return Predictability. Unpublished Manuscript.

Kritzman, M., Page, S., & Turkington, D. (2012). Regime Shifts: Implications for Dynamic Strategies. Financial Analysts Journal, 68(3).

Malmendier, U. (2005). Roman Shares. In W. Goetzmann, & G. Rouwenhorst, *The Origins of Value. The Financial Innovations that Creat Modern Capital Markets* (pp. 31-42 , 361-365). Oxford University Press.

Meligkotsidou, L., & Dellaportas, P. (2011). Forecasting with Non-homogeneous Hidden Markov Models. *Statistics and Computing*, 21: 439-449.

Myers, C., & Rabiner, L. (1981). A Comparative Study of Several Dynamic Time-Warping Algorithms for Connected-Word Recognition. *The Bell System Technical Journal*, 1389-1409.

Pagan, A., & Schwert, G. (1990). Alternative Models for Conditional Stock Volatility. *Journal of Econometrics*, 45: 267-290.

Rabiner, L. (1989). A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77: 257-286.

Rydén,T., Teräsvirta, T., & Åsbrink, S. (1990). Stylized Facts of Daily Return Series and the Hidden Markov Model. *Journal ofApplied Econometrics*, 45(3): 217-244.

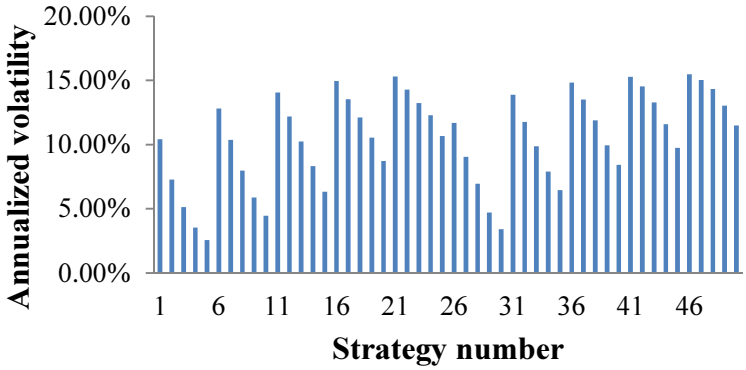
Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2): 461-464.

Zhang, Y. (2001). Prediction of Financial Time Series with Hidden Markov Models. Unpublished master dissertation, Shadong University, China.

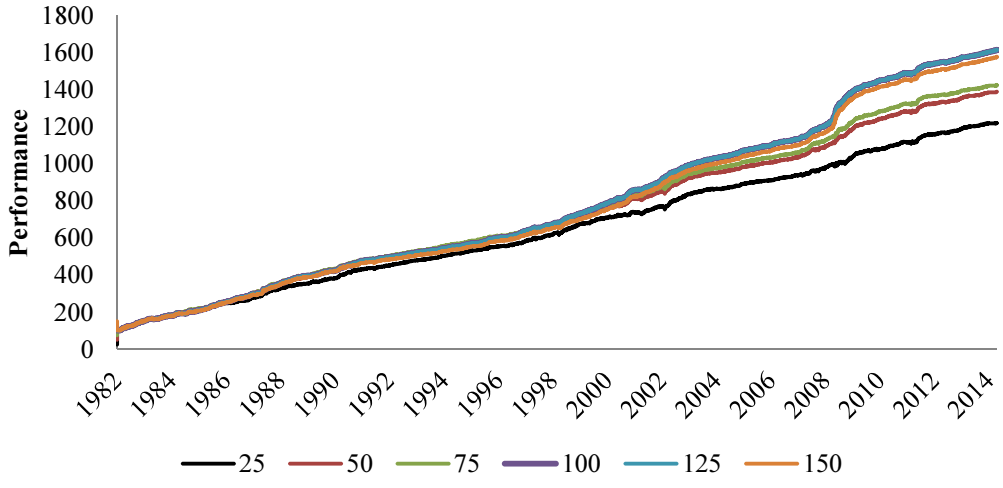
6. Appendix

Appendix 1 - Annualized return per strategy

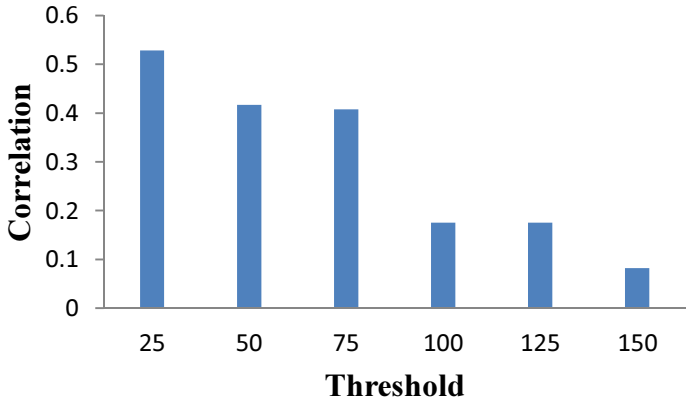
Appendix 2 - Annualized volatility per strategy



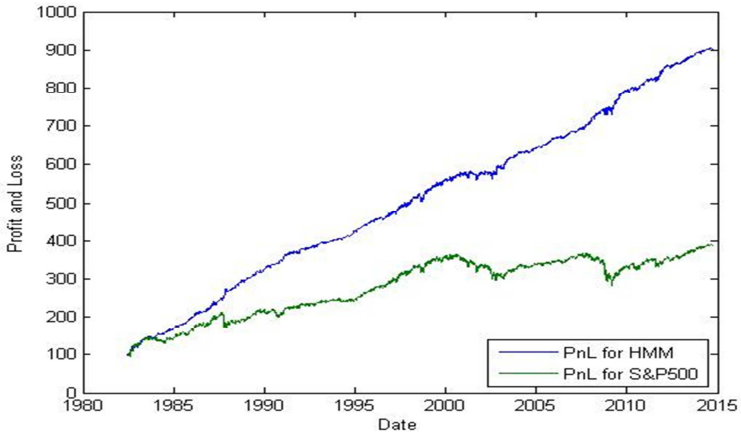
Appendix 3 – Performance against several thresholds for the number of training observations



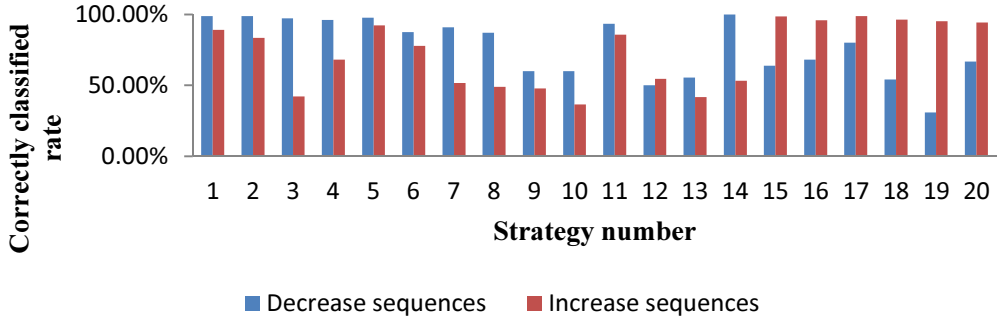
Appendix 4 – Correlation between the market and the strategy for a set of thresholds for number of training observations



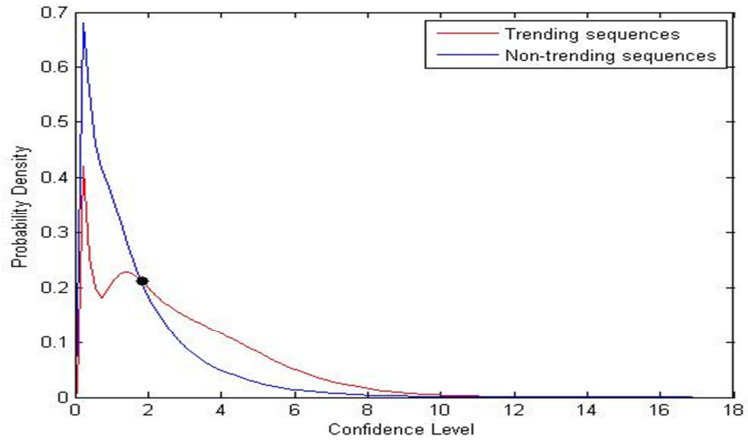
Appendix 5 – Profit and Loss (PnL) plot for HMM strategy and S&P500 Index



Appendix 6 - Rate of correctly classified sequences, for both increase and decrease sequences



Appendix 7 – Probability density function of the confidence for both increase and decrease sequences



Appendix 8 – Performance metrics for garbage model strategies

	Garbage model 1	Garbage model 2	PDF model	S&P500 Index
Annualized return	3.59%	3.10%	2.75%	8.96%
Annualized volatility	13.19%	13.04%	13.78%	18.17%
RR	0.27	0.24	0.20	0.49
Correlation to the market	-0.72%	23.69%	-0.38%	100.00%
% of positive days	0.53	0.53	0.53	0.54
% of positive months	0.59	0.57	0.63	0.77
% of positive years	0.64	0.55	0.61	0.61
Kurtosis	88.62	92.99	76.86	30.59
Skewness	-2.47	-2.59	-2.23	-1.22
Maximum Drawdown	49.36%	53.02%	47.45%	22.88%

Appendix 9 – Average state number per garbage model according to AIC